D 8

```
BBBBBBBBBBBB         AAAAAAAAA     DDDDDDDDDDDD
BBBBBBBBBBBB         AAAAAAAAA     DDDDDDDDDDDD
BBBBBBBBBBBB         AAAAAAAAA     DDDDDDDDDDDD
BBB         BBB  AAA          AAA  DDD         DDD
BBB         BBB  AAA          AAA  DDD         DDD
BBB         BBB  AAA          AAA  DDD         DDD
BBB         BBB  AAA          AAA  DDD         DDD
BBB         BBB  AAA          AAA  DDD         DDD
BBB         BBB  AAA          AAA  DDD         DDD
BBBBBBBBBBBB     AAA          AAA  DDD         DDD
BBBBBBBBBBBB     AAA          AAA  DDD         DDD
BBBBBBBBBBBB     AAA          AAA  DDD         DDD
BBB         BBB  AAAAAAAAAAAAAAAA  DDD         DDD
BBB         BBB  AAAAAAAAAAAAAAAA  DDD         DDD
BBB         BBB  AAAAAAAAAAAAAAAA  DDD         DDD
BBB         BBB  AAA          AAA  DDD         DDD
BBB         BBB  AAA          AAA  DDD         DDD
BBB         BBB  AAA          AAA  DDD         DDD
BBBBBBBBBBBB     AAA          AAA  DDDDDDDDDDDD
BBBBBBBBBBBB     AAA          AAA  DDDDDDDDDDDD
BBBBBBBBBBBB     AAA          AAA  DDDDDDDDDDDD
```

```
                                                  D 9
**FILE**ID**BADREPORT




BBBBBBBB      AAAAAA    DDDDDDDD    RRRRRRRR    EEEEEEEEEE  PPPPPPPP      OOOOOO    RRRRRRRR    TTTTTTTTTT
BBBBBBBB      AAAAAA    DDDDDDDD    RRRRRRRR    EEEEEEEEEE  PPPPPPPP      OOOOOO    RRRRRRRR    TTTTTTTTTT
BB      BB  AA      AA  DD      DD  RR      RR  EE          PP      PP  OO      OO  RR      RR      TT
BB      BB  AA      AA  DD      DD  RR      RR  EE          PP      PP  OO      OO  RR      RR      TT
BB      BB  AA      AA  DD      DD  RR      RR  EE          PP      PP  OO      OO  RR      RR      TT
BB      BB  AA      AA  DD      DD  RR      RR  EE          PP      PP  OO      OO  RR      RR      TT
BBBBBBBB    AA      AA  DD      DD  RRRRRRRR    EEEEEEEE    PPPPPPPP    OO      OO  RRRRRRRR        TT
BBBBBBBB    AA      AA  DD      DD  RRRRRRRR    EEEEEEEE    PPPPPPPP    OO      OO  RRRRRRRR        TT
BB      BB  AAAAAAAAAA  DD      DD  RR  RR      EE          PP          OO      OO  RR  RR          TT
BB      BB  AAAAAAAAAA  DD      DD  RR  RR      EE          PP          OO      OO  RR  RR          TT
BB      BB  AA      AA  DD      DD  RR    RR    EE          PP          OO      OO  RR    RR        TT           ....
BB      BB  AA      AA  DD      DD  RR    RR    EE          PP          OO      OO  RR    RR        TT           ....
BBBBBBBB    AA      AA  DDDDDDDD    RR      RR  EEEEEEEEEE  PP            OOOOOO    RR      RR      TT           ....
BBBBBBBB    AA      AA  DDDDDDDD    RR      RR  EEEEEEEEEE  PP            OOOOOO    RR      RR      TT           ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;      1     0001  0 MODULE badreport          (%TITLE 'Analyze/Media Report Generation Module'
;      2     0002  0                           IDENT = 'V04-000') =
;      3     0003  1 BEGIN
;      4     0004  1 !
;      5     0005  1 !****************************************************************************
;      6     0006  1 !*                                                                          *
;      7     0007  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
;      8     0008  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
;      9     0009  1 !*  ALL RIGHTS RESERVED.                                                    *
;     10     0010  1 !*                                                                          *
;     11     0011  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;     12     0012  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
;     13     0013  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;     14     0014  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;     15     0015  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;     16     0016  1 !*  TRANSFERRED.                                                            *
;     17     0017  1 !*                                                                          *
;     18     0018  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;     19     0019  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;     20     0020  1 !*  CORPORATION.                                                            *
;     21     0021  1 !*                                                                          *
;     22     0022  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;     23     0023  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;     24     0024  1 !*                                                                          *
;     25     0025  1 !*                                                                          *
;     26     0026  1 !****************************************************************************
;     27     0027  1 !
;     28     0028  1 !
;     29     0029  1 !++
;     30     0030  1 !
;     31     0031  1 ! Facility:
;     32     0032  1 !
;     33     0033  1 !       Analyze/Media
;     34     0034  1 !
;     35     0035  1 ! Abstract:
;     36     0036  1 !
;     37     0037  1 !       This module contains the routines necessary to produce a listing of the
;     38     0038  1 !       known bad blocks, which have been recorded in the MDBSF and SDBSF.
;     39     0039  1 !
;     40     0040  1 ! Author:
;     41     0041  1 !
;     42     0042  1 !       Michael T. Rhodes,              Creation Date: July, 1982
;     43     0043  1 !
;     44     0044  1 ! Modified By:
;     45     0045  1 !
;     46     0046  1 !       V03-001 MTR0008         Michael T. Rhodes               22-Jul-1983
;     47     0047  1 !               Covert RMS error reporting mechanism from local signalling to
;     48     0048  1 !               the common UTIL$REPORT_IO_ERROR routine.
;     49     0049  1 !
;     50     0050  1 !--
;    52   0051  1 %SBTTL  'Declarations'
;    53   0052  1 !
;    54   0053  1 ! Include Files:
;    55   0054  1 !
;    56   0055  1 REQUIRE 'lib$:baddef';                                  ! Define BAD's structures etc.
;    57   0175  1 LIBRARY 'SYS$LIBRARY:LIB';                              ! Define VMS structures etc.
;    58   0176  1
;    59   0177  1 !
;    60   0178  1 ! Table of Contents:
;    61   0179  1 !
;    62   0180  1 FORWARD ROUTINE
;    63   0181  1     bad$prepare_report   : NOVALUE,                     ! Open output file and connect data streams.
;    64   0182  1     bad$produce_report   : NOVALUE,                     ! Produce the report.
;    65   0183  1     bad$close_files      : NOVALUE,                     ! Close the output files.
;    66   0184  1     new_page             : NOVALUE,                     ! Generate a new page.
;    67   0185  1     put_line             : NOVALUE,                     ! Write a new line to the output file.
;    68   0186  1     format_line          : NOVALUE,                     ! Format the current line.
;    69   0187  1     get_entry,                                          ! Extract and convert a last track entry to longword
;    70   0188  1     list_last_track      : NOVALUE,                     ! List the last track bad block buffers (MDBSF and S
;    71   0189  1     list_non_lsttrk      : NOVALUE;                     ! List the non last track bad block buffer SDBSF.
;    72   0190  1
;    73   0191  1 !
;    74   0192  1 ! External References:
;    75   0193  1 !
;    76   0194  1 EXTERNAL ROUTINE
;    77   0195  1     bad$cvt_ltk_long     : ADDRESSING_MODE (GENERAL),   ! Convert last track entry to longwords.
;    78   0196  1     bad$cvt_nlt_long     : ADDRESSING_MODE (GENERAL),   ! Convert non last track entry to longwords.
;    79   0197  1     lib$lp_lines         : ADDRESSING_MODE (GENERAL),   ! Obtain line printer lines per page characteristic.
;    80   0198  1     util$report_io_error: ADDRESSING_MODE (GENERAL);    ! Report RMS IO errors from FAB or RAB as appropriat
;    81   0199  1
;    82   0200  1 EXTERNAL
;    83   0201  1     bad$ga_comnd_line    : $BBLOCK [dsc$c_s_bln],       ! Command line descriptor.
;    84   0202  1     bad$gl_context       : BITVECTOR [32],              ! Context bits.
;    85   0203  1     bad$ga_device        : $BBLOCK [dsc$c_s_bln],       ! Device name descriptor.
;    86   0204  1     bad$gq_devnam        : $BBLOCK [dsc$c_s_bln],       ! Device name descriptor.
;    87   0205  1     bad$ga_filespec      : $BBLOCK [dsc$c_s_bln],       ! File name descriptor.
;    88   0206  1     bad$gl_maxblock,                                    ! Total number of blocks on the device.
;    89   0207  1     bad$ga_mdbsf         : REF $BBLOCK,                 ! Address of the MDBSF.
;    90   0208  1     bad$ga_sdbsf         : REF $BBLOCK,                 ! Address of the SDBSF.
;    91   0209  1     bad$gl_serialnum,                                   ! Serial number for device.
;    92   0210  1     bad$gl_status;                                      ! Global status.
;    93   0211  1
;    94   0212  1 !
;    95   0213  1 ! Define message codes...
;    96   0214  1 !
;    97   0215  1 EXTERNAL LITERAL
;    98   0216  1     bad$_blk0bad,                                       ! Block 0 is bad.
;    99   0217  1     bad$_closeout,                                      ! Error closing output.
;   100   0218  1     bad$_fao,                                           ! Error formatting data with FAO.
;   101   0219  1     bad$_getmsg,                                        ! Error obtaining message with GETMSG.
;   102   0220  1     bad$_heading1,                                      ! Heading line 1.
;   103   0221  1     bad$_heading2,                                      ! Heading line 2.
;   104   0222  1     bad$_lstmdbsf,                                      ! MDBSF/SDBSF heading line.
;   105   0223  1     bad$_lstmdbs1,                                      ! MDBSF/SDBSF sub heading line.
;   106   0224  1     bad$_lstmdbs2,                                      ! MDBSF/SDBSF sub heading line.
;   107   0225  1     bad$_lstmdbs3,                                      ! MDBSF/SDBSF data line.
;   108   0226  1     bad$_lstmdbs4,                                      ! MDBSF/SDBSF count line.
;  109       0227  1       bad$_lstsdbsf,                          ! SDBSF heading line.
;  110       0228  1       bad$_lstsdbs1,                          ! SDBSF sub heading line.
;  111       0229  1       bad$_lstsdbs2,                          ! SDBSF sub heading line.
;  112       0230  1       bad$_lstsdbs3,                          ! SDBSF data line.
;  113       0231  1       bad$_lsttotbk,                          ! Total bad blocks line.
;  114       0232  1       bad$_lststring,                         ! List any ASCID string.
;  115       0233  1       bad$_openout,                           ! Error opening output file.
;  116       0234  1       bad$_writeerr;                          ! Write error.
;  117       0235  1
;  118       0236  1 !
;  119       0237  1 ! Private Storage:
;  120       0238  1 !
;  121       0239  1 OWN
;  122       0240  1       report_fab : $FAB_DECL,                 ! Address of the report file FAB.
;  123       0241  1       report_rab : $RAB_DECL,                 ! Address of the report file RAB.
;  124       0242  1       line_count,                             ! Number of lines remaining on a page.
;  125       0243  1       mdbsf_count,                            ! Number of entries in the MDBSF.
;  126       0244  1       page_number,                            ! Page counter.
;  127       0245  1       sdbsf_count;                            ! Number of enteries in the SDBSF.
;  128       0246  1
;  129       0247  1 LITERAL
;  130       0248  1       formfeed = 12;
;  131       0249  1
;  133          0250   1 %SBTTL  'bad$prepare_report -- Prepare Report File'
;  134          0251   1 GLOBAL ROUTINE bad$prepare_report : NOVALUE =
;  135          0252   1 !++
;  136          0253   1 !
;  137          0254   1 ! Functional Description:
;  138          0255   1 !
;  139          0256   1 !     This routine is responsible for obtaining the file spec for an output
;  140          0257   1 !     listing file and establishing the data stream between the program and
;  141          0258   1 !     that file.
;  142          0259   1 !
;  143          0260   1 ! Implicit Inputs:
;  144          0261   1 !
;  145          0262   1 !     bad$ga_filespec     - Address of the file spec descriptor.
;  146          0263   1 !
;  147          0264   1 ! Side Effects:
;  148          0265   1 !
;  149          0266   1 !     The output listing file is opened.
;  150          0267   1 !
;  151          0268   1 !--
;  152          0269   2 BEGIN
;  153        P 0270   2 $FAB_INIT  (FAB=report_fab,                               ! Initialize the FAB.
;  154        P 0271   2             DNM='ANALYZE.ANL',
;  155        P 0272   2             FAC=put,
;  156        P 0273   2             FOP=sqo,
;  157        P 0274   2             ORG=seq,
;  158        P 0275   2             RAT=cr,
;  159          0276   2             RFM=var);
;  160          0277   2
;  161        P 0278   2 $RAB_INIT  (RAB=report_rab,                               ! Initialize the RAB.
;  162        P 0279   2             FAB=report_fab,
;  163          0280   2             RAC=seq);
;  164          0281   2
;  165          0282   2 IF .bad$ga_filespec [dsc$w_length] NEQ 0                  ! If the user supplied
;  166          0283   2 THEN                                                      ! a filespec, use it.
;  167          0284   3     BEGIN
;  168          0285   3     report_fab [fab$l_fna] = .bad$ga_filespec [dsc$a_pointer];
;  169          0286   3     report_fab [fab$b_fns] = .bad$ga_filespec [dsc$w_length];
;  170          0287   2     END;
;  171          0288   2
;  172          0289   2 report_fab [fab$l_ctx] = bad$_openout;                    ! Pass the error code
;  173          0290   2 report_rab [rab$l_ctx] = bad$_openout;                    ! for processing context.
;  174          0291   2 $CREATE (FAB = report_fab, ERR = util$report_io_error);  ! Create the output file.
;  175          0292   2 $CONNECT (RAB = report_rab, ERR = util$report_io_error);! Connect the data stream.
;  176          0293   2
;  177          0294   2 page_number = 0;                                          ! Initialize the page number.
;  178          0295   2
;  179          0296   1 END;     ! of GLOBAL ROUTINE bad$prepare_report


                                                            .TITLE  BADREPORT Analyze/Media Report Generation Modul
;                                                                   e
                                                            .IDENT  \V04-000\

                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2

            4C  4E  41  2E  45  5A  59  4C  41  4E  41  00000 P.AAA:  .ASCII  \ANALYZE.ANL\                        ;
```

```
                                                                     .PSECT  $OWN$,NOEXE,2

                                                 00000 REPORT_FAB:
                                                                     .BLKB   80
                                                 00050 REPORT_RAB:
                                                                     .BLKB   68
                                                 00094 LINE_COUNT:
                                                                     .BLKB   4
                                                 00098 MDBSF_COUNT:
                                                                     .BLKB   4
                                                 0009C PAGE_NUMBER:
                                                                     .BLKB   4
                                                 000A0 SDBSF_COUNT:
                                                                     .BLKB   4

                                                       $RMS_PTR=             REPORT_FAB
                                                       $RMS_PTR=             REPORT_RAB
                                                                     .EXTRN  BAD$CVT_LTR_LONG
                                                                     .EXTRN  BAD$CVT_NLT_LONG
                                                                     .EXTRN  LIB$LP_LINES, UTIL$REPORT_IO_ERROR
                                                                     .EXTRN  BAD$GA_COMND_LINE
                                                                     .EXTRN  BAD$GL_CONTEXT, BAD$GA_DEVICE
                                                                     .EXTRN  BAD$GQ_DEVNAM, BAD$GA_FILESPEC
                                                                     .EXTRN  BAD$GL_MAXBLOCK
                                                                     .EXTRN  BAD$GA_MDBSF, BAD$GA_SDBSF
                                                                     .EXTRN  BAD$GL_SERIALNUM
                                                                     .EXTRN  BAD$GL_STATUS, BAD$_BLK0BAD
                                                                     .EXTRN  BAD$_CLOSEOUT, BAD$_FAO
                                                                     .EXTRN  BAD$_GETMSG, BAD$_HEADING1
                                                                     .EXTRN  BAD$_HEADING2, BAD$_LSTMDBSF
                                                                     .EXTRN  BAD$_LSTMDBS1, BAD$_LSTMDBS2
                                                                     .EXTRN  BAD$_LSTMDBS3, BAD$_LSTMDBS4
                                                                     .EXTRN  BAD$_LSTSDBSF, BAD$_LSTSDBS1
                                                                     .EXTRN  BAD$_LSTSDBS2, BAD$_LSTSDBS3
                                                                     .EXTRN  BAD$_LSTTOTBK, BAD$_LSTSTRING
                                                                     .EXTRN  BAD$_OPENOUT, BAD$_WRITEERR
                                                                     .EXTRN  SYS$CREATE, SYS$CONNECT

                                                                     .PSECT  $CODE$,NOWRT,2

                                            01FC 00000               .ENTRY  BAD$PREPARE_REPORT, Save R2,R3,R4,R5,R6,R7,-; 0251
                                                                             R8                                          :
                         58 00000000G    00   9E 00002               MOVAB   UTIL$REPORT_IO_ERROR, R8                    :
                         57 00000000G    8F   D0 00009               MOVL    #BAD$_OPENOUT, R7                           :
                         56     0000'    CF   9E 00010               MOVAB   $RMS_PTR, R6                                :
 0050   8F          00   6E              00   2C 00015               MOVC5   #0, (SP), #0, #80, $RMS_PTR                 : 0276
                                         66      0001C                                                                   :
                         66     5003     8F   B0 0001D               MOVW    #20483, $RMS_PTR                            :
                   04    A6       40     8F   9A 00022               MOVZBL  #64, $RMS_PTR+4                             :
                   16    A6              01   90 00027               MOVB    #1, $RMS_PTR+22                             :
                   1D    A6     0200     8F   B0 0002B               MOVW    #512, $RMS_PTR+29                           :
                   1F    A6              02   90 00031               MOVB    #2, $RMS_PTR+31                             :
                   30    A6     0000'    CF   9E 00035               MOVAB   P.AAA, $RMS_PTR+48                          :
                   35    A6              0B   90 0003B               MOVB    #11, $RMS_PTR+53                            :
 0044   8F          00   6E              00   2C 0003F               MOVC5   #0, (SP), #0, #68, $RMS_PTR                 : 0280
                                  50     A6      00046                                                                   :
                   50    A6     4401     8F   B0 00048               MOVW    #17409, $RMS_PTR                            :
```

```
                                 6E   A6   94 0004E           CLRB     $RMS_PTR+30                              ;
               008C    C6              66   9E 00051           MOVAB    REPORT_FAB, $RMS_PTR+60                  ;
                       50     0000G   CF   3C 00056           MOVZWL   BAD$GA_FILESPEC, R0                      ; 0282
                                      0A   13 0005B           BEQL     1$                                       ;
                 2C    A6     0000G   CF   D0 0005D           MOVL     BAD$GA_FILESPEC+4, REPORT_FAB+44         ; 0285
                 34    A6              50   90 00063           MOVB     R0, REPORT_FAB+52                        ; 0286
                 18    A6              57   D0 00067 1$:       MOVL     R7, REPORT_FAB+24                        ; 0289
                 68    A6              57   D0 0006B           MOVL     R7, REPORT_RAB+24                        ; 0290
                              0140    8F   BB 0006F           PUSHR    #^M<R6,R8>                               ; 0291
          00000000G    00              02   FB 00073           CALLS    #2, SYS$CREATE                           ;
                                      58   DD 0007A           PUSHL    R8                                       ; 0292
                                50    A6   9F 0007C           PUSHAB   REPORT_RAB                               ;
          00000000G    00              02   FB 0007F           CALLS    #2, SYS$CONNECT                          ;
                              009C    C6   D4 00086           CLRL     PAGE_NUMBER                              ; 0294
                                           04 0008A           RET                                               ; 0296

; Routine Size:  139 bytes,    Routine Base:  $CODE$ + 0000


;  180         0297  1
```

```
1 %SBTTL  'new_page -- Generate a new page in the listing file'
1 ROUTINE new_page : NOVALUE =
1 !++
1 !
1 ! Functional Description:
1 !
1 !    The routine generates a new page in the listing file, resets the line
1 !    counter, and generates the heading lines.
1 !
1 ! Implicit Inputs:
1 !
1 !    Global data.
1 !
1 ! Side Effects:
1 !
1 !    A new page is generated with headings and the line count is reset.
1 !
1 !--
2 BEGIN
2
2 line_count = LIB$LP_LINES() - 9;                        ! Set the number of lines
2                                                         ! remaining on the page.
2 put_line ($descriptor (%CHAR (formfeed)));              ! Skip to top of page.
2
2 page_number = .page_number + 1;                         ! Increment the page number.
2 format_line (bad$_heading1, 0, .page_number);           ! Generate the headings.
2 format_line (bad$_heading2, bad$ga_device, bad$gq_devnam, .bad$gl_serialnum);
2 put_line ();                                            ! Skip a couple of lines.
2 put_line ();
2
2 IF .bad$gl_context [ctx_v_ltdevice]
2 THEN
3     BEGIN                                               ! Write the last track
3     format_line (bad$_lstmdbs1);                        ! device sub headings.
3     put_line ();
3     format_line (bad$_lstmdbs2);
3     END
2 ELSE
3     BEGIN                                               ! Write the non last track
3     format_line (bad$_lstsdbs1);                        ! device sub headings.
3     put_line ();
3     format_line (bad$_lstsdbs2);
2     END;
2
1 END;    ! of ROUTINE new_page


                                                     .PSECT  $PLIT$,NOWRT,NOEXE,2

                                  0C  0000B P.AAC:   .ASCII  <12>                                             ;
                            00000001  0000C P.AAB:   .LONG   1                                                ;
                            00000000' 00010          .ADDRESS P.AAC                                           ;


                                                     .PSECT  $CODE$,NOWRT,2
```

```
                                     000C 00000 NEW_PAGE:
                                                        .WORD   Save R2,R3                                     ; 0299
                    53      0000V  CF  9E 00002         MOVAB   FORMAT_LINE, R3                                :
                    52      0000V  CF  9E 00007         MOVAB   PUT_LINE, R2                                   :
          00000000G 00             00  FB 0000C         CALLS   #0, LIB$LP_LINES                               ; 0318
              0000' CF        F7   A0  9E 00013         MOVAB   -9(R0), LINE_COUNT                             :
                            0000'  CF  9F 00019         PUSHAB  P.AAB                                          ; 0320
                    62             01  FB 0001D         CALLS   #1, PUT_LINE                                   :
                            0000'  CF  D6 00020         INCL    PAGE_NUMBER                                    ; 0322
                            0000'  CF  DD 00024         PUSHL   PAGE_NUMBER                                    ; 0323
                                   7E  D4 00028         CLRL    -(SP)                                          :
                        00000000G  8F  DD 0002A         PUSHL   #BAD$_HEADING1                                 :
                    63             03  FB 00030         CALLS   #3, FORMAT_LINE                                :
                            0000G  CF  DD 00033         PUSHL   BAD$GL_SERIALNUM                               ; 0324
                            0000G  CF  9F 00037         PUSHAB  BAD$GQ_DEVNAM                                  :
                            0000G  CF  9F 0003B         PUSHAB  BAD$GA_DEVICE                                  :
                        00000000G  8F  DD 0003F         PUSHL   #BAD$_HEADING2                                 :
                    63             04  FB 00045         CALLS   #4, FORMAT_LINE                                :
                    62             00  FB 00048         CALLS   #0, PUT_LINE                                   ; 0325
                    62             00  FB 0004B         CALLS   #0, PUT_LINE                                   ; 0326
                    14      0000G  CF  E9 0004E         BLBC    BAD$GL_CONTEXT+1, 1$                           ; 0328
                        00000000G  8F  DD 00053         PUSHL   #BAD$_LSTMDBS1                                 ; 0331
                    63             01  FB 00059         CALLS   #1, FORMAT_LINE                                :
                    62             00  FB 0005C         CALLS   #0, PUT_LINE                                   ; 0332
                        00000000G  8F  DD 0005F         PUSHL   #BAD$_LSTMDBS2                                 ; 0333
                                   12  11 00065         BRB     2$                                             :
                        00000000G  8F  DD 00067 1$:     PUSHL   #BAD$_LSTSDBS1                                 ; 0337
                    63             01  FB 0006D         CALLS   #1, FORMAT_LINE                                :
                    62             00  FB 00070         CALLS   #0, PUT_LINE                                   ; 0338
                        00000000G  8F  DD 00073         PUSHL   #BAD$_LSTSDBS2                                 ; 0339
                    63             01  FB 00079 2$:     CALLS   #1, FORMAT_LINE                                :
                                       04 0007C         RET                                                    ; 0342

; Routine Size:  125 bytes,    Routine Base:  $CODE$ + 008B


;  227          0343  1
;   229       0344  1 %SBTTL  'put_line -- Write the current line to the output file'
;   230       0345  1 ROUTINE put_line (line) : NOVALUE =
;   231       0346  1 !++
;   232       0347  1 !
;   233       0348  1 ! Functional Description:
;   234       0349  1 !
;   235       0350  1 !     This routine writes the specified line to the output file.
;   236       0351  1 !     Once the line is written the line counter is decremented and if it
;   237       0352  1 !     is LEQ 0 then a new page is generated.
;   238       0353  1 !
;   239       0354  1 ! Inputs:
;   240       0355  1 !
;   241       0356  1 !     line         adr      address of the string descriptor for the line.
;   242       0357  1 !
;   243       0358  1 ! Implicit Outputs:
;   244       0359  1 !
;   245       0360  1 !     The line has been written to the output file.
;   246       0361  1 !
;   247       0362  1 !--
;   248       0363  2 BEGIN
;   249       0364  2 BIND
;   250       0365  2     line_desc = line : REF $BBLOCK;
;   251       0366  2
;   252       0367  2 BUILTIN
;   253       0368  2     NULLPARAMETER;
;   254       0369  2
;   255       0370  2 IF NULLPARAMETER(1)                                   ! If no line descriptor
;   256       0371  2 THEN                                                  ! is specified, then we
;   257       0372  2     report_rab [rab$w_rsz] = 0                        ! generate a blank line.
;   258       0373  2 ELSE
;   259       0374  3     BEGIN                                             ! Set the size and address
;   260       0375  3     report_rab [rab$w_rsz] = .line_desc [dsc$w_length]; ! of the line, into the
;   261       0376  3     report_rab [rab$l_rbf] = .line_desc [dsc$a_pointer];! RAB's record control.
;   262       0377  2     END;
;   263       0378  2
;   264       0379  2 report_rab [rab$l_ctx] = bad$_writeerr;               ! Pass error code.
;   265       0380  2 $PUT (RAB = report_rab, ERR = util$report_io_error);   ! Write the line.
;   266       0381  2
;   267       0382  2 line_count = .line_count - 1;                         ! Decrement the line count.
;   268       0383  2 IF .line_count LEQ 0                                  ! Time for a new page?
;   269       0384  2 THEN
;   270       0385  2     new_page ();                                      ! Yes -- generate a new page.
;   271       0386  2
;   272       0387  1 END;    ! of ROUTINE put_line


                                                  .EXTRN  SYS$PUT

                                   0004 00000 PUT_LINE:
                                                  .WORD   Save R2                                    ; 0345
                 52      0000'  CF  9E 00002      MOVAB   REPORT_RAB+34, R2                          ;
                                6C  95 00007      TSTB    (AP)                                       ; 0370
                                05  13 00009      BEQL    1$                                         ;
                            04  AC  D5 0000B      TSTL    4(AP)                                      ;
                                04  12 0000E      BNEQ    2$                                         ;
                                62  B4 00010 1$:  CLRW    REPORT_RAB+34                              ; 0372
```

```
                                        0C  11 00012       BRB     3$                                            ;
                    50          04      AC  D0 00014  2$:  MOVL    LINE_DESC, R0                                 ; 0375
                    62                  60  B0 00018       MOVW    (R0), REPORT_RAB+34                           ;
              06    A2          04      A0  D0 0001B       MOVL    4(R0), REPORT_RAB+40                          ; 0376
              F6    A2   00000000G      8F  D0 00020  3$:  MOVL    #BAD$_WRITEERR, REPORT_RAB+24                 ; 0379
                         00000000G      00  9F 00028       PUSHAB  UTIL$REPORT_IO_ERROR                          ; 0380
                                DE      A2  9F 0002E       PUSHAB  REPORT_RAB                                    ;
       00000000G    00                  02  FB 00031       CALLS   #2, SYS$PUT                                   ;
                    05          22      A2  F5 00038       SOBGTR  LINE_COUNT, 4$                                ; 0382
            FF42    CF                  00  FB 0003C       CALLS   #0, NEW_PAGE                                  ; 0385
                                            04 00041  4$:  RET                                                   ; 0387

; Routine Size:  66 bytes,    Routine Base:  $CODE$ + 0108


;  273         0388  1
```

```
  1 %SBTTL  'format_line -- Format the output line with supplied data'
  1 ROUTINE format_line (message, fao_args) : NOVALUE =
  1 !++
  1 !
  1 ! Functional Description:
  1 !
  1 !    This routine calls the system services $GETMSG and $FAOL, which will
  1 !    allow us to obtain the base control string and format the supplied text
  1 !    into a printable record.
  1 !
  1 ! Inputs:
  1 !
  1 !    message    val      the message number used to obtain the control text.
  1 !    fao_args   adr      the address of an fao argument list.
  1 !
  1 ! Implicit Outputs:
  1 !
  1 !    Once the string has been formatted, we call put_line() to write the
  1 !    record to the output file.
  1 !
  1 !--
  2 BEGIN
  2 LOCAL
  2     ctr_buffer  : $BBLOCK [132],                          ! Control string buffer.
  2     ctr_buf_dsc : $BBLOCK [dsc$c_s_bln],                  ! Control string buffer descriptor.
  2
  2     res_buffer  : $BBLOCK [132],                          ! Result buffer.
  2     res_buf_dsc : $BBLOCK [dsc$c_s_bln];                  ! Result buffer descriptor.
  2
  2 CH$FILL (0, dsc$c_s_bln, ctr_buf_dsc);                    ! Initialize the descriptor.
  2 CH$FILL (0, dsc$c_s_bln, res_buf_dsc);
  2 ctr_buf_dsc [dsc$w_length] = res_buf_dsc [dsc$w_length] = 132; ! Set the size.
  2 ctr_buf_dsc [dsc$b_class] = dsc$k_class_d;                ! Initialize the descriptor class.
  2 res_buf_dsc [dsc$b_class] = dsc$k_class_d;
  2 ctr_buf_dsc [dsc$a_pointer]= ctr_buffer;                  ! Set the buffer addresses.
  2 res_buf_dsc [dsc$a_pointer]= res_buffer;
  2
P 3 IF NOT (bad$gl_status = $GETMSG (MSGID  = .message,      ! Fetch the control string
P 3                                  MSGLEN = ctr_buf_dsc,    ! with text ONLY.
P 3                                  BUFADR = ctr_buf_dsc,
  3                                  FLAGS  = %B'0001'))
  2 THEN
  2     SIGNAL (bad$_getmsg, 0, .bad$gl_status);
  2
P 3 IF NOT (bad$gl_status = $FAOL (CTRSTR = ctr_buf_dsc,     ! Format the text with
P 3                                OUTLEN = res_buf_dsc,      ! the control string,
P 3                                OUTBUF = res_buf_dsc,      ! leaving us the
  3                                PRMLST = fao_args))        ! printable result.
  2 THEN
  2     SIGNAL (bad$_fao, 0, .bad$gl_status);
  2
  2 put_line (res_buf_dsc);                                   ! Write result to the output file.
  2
  1 END;     ! of ROUTINE format_line
```

```
                                                             .EXTRN  SYS$GETMSG, SYS$FAOL

                                          00FC 00000 FORMAT_LINE:
                                                             .WORD   Save R2,R3,R4,R5,R6,R7            ; 0390
                          57 00000000G  00  9E 00002         MOVAB   LIB$SIGNAL, R7                    ;
                          56     0000G  CF  9E 00009         MOVAB   BAD$GL_STATUS, R6                 ;
                          5E      FEE8  CE  9E 0000E         MOVAB   -280(SP), SP                      ;
  08        00            6E            00  2C 00013         MOVC5   #0, (SP), #0, #8, CTR_BUF_DSC     ; 0418
                                  FF74  CD     00018                                                   ;
  08        00            6E            00  2C 0001B         MOVC5   #0, (SP), #0, #8, RES_BUF_DSC     ; 0419
                                        6E     00020                                                   ;
                          6E        84  8F  9B 00021         MOVZBW  #132, RES_BUF_DSC                 ; 0420
                     FF74 CD        84  8F  9B 00025         MOVZBW  #132, CTR_BUF_DSC                 ;
                     FF77 CD            02  90 0002B         MOVB    #2, CTR_BUF_DSC+3                 ; 0421
                       03 AE            02  90 00030         MOVB    #2, RES_BUF_DSC+3                 ; 0422
                     FF78 CD      FF7C  CD  9E 00034         MOVAB   CTR_BUFFER, CTR_BUF_DSC+4         ; 0423
                       04 AE        08  AE  9E 0003B         MOVAB   RES_BUFFER, RES_BUF_DSC+4         ; 0424
                          7E            01  7D 00040         MOVQ    #1, -(SP)                         ; 0429
                                  FF74  CD  9F 00043         PUSHAB  CTR_BUF_DSC                       ;
                                  FF74  CD  9F 00047         PUSHAB  CTR_BUF_DSC                       ;
                                    04  AC  DD 0004B         PUSHL   MESSAGE                           ;
                00000000G 00            05  FB 0004E         CALLS   #5, SYS$GETMSG                    ;
                          66            50  D0 00055         MOVL    R0, BAD$GL_STATUS                 ;
                          0D            50  E8 00058         BLBS    R0, 1$                            ;
                                        66  DD 0005B         PUSHL   BAD$GL_STATUS                     ; 0431
                                        7E  D4 0005D         CLRL    -(SP)                             ;
                             00000000G  8F  DD 0005F         PUSHL   #BAD$_GETMSG                      ;
                          67            03  FB 00065         CALLS   #3, LIB$SIGNAL                    ;
                                    08  AC  9F 00068 1$:     PUSHAB  FAO_ARGS                          ; 0436
                                    04  AE  9F 0006B         PUSHAB  RES_BUF_DSC                       ;
                                    08  AE  9F 0006E         PUSHAB  RES_BUF_DSC                       ;
                                  FF74  CD  9F 00071         PUSHAB  CTR_BUF_DSC                       ;
                00000000G 00            04  FB 00075         CALLS   #4, SYS$FAOL                      ;
                          66            50  D0 0007C         MOVL    R0, BAD$GL_STATUS                 ;
                          0D            50  E8 0007F         BLBS    R0, 2$                            ;
                                        66  DD 00082         PUSHL   BAD$GL_STATUS                     ; 0438
                                        7E  D4 00084         CLRL    -(SP)                             ;
                             00000000G  8F  DD 00086         PUSHL   #BAD$_FAO                         ;
                          67            03  FB 0008C         CALLS   #3, LIB$SIGNAL                    ;
                                        5E  DD 0008F 2$:     PUSHL   SP                                ; 0440
                     FF28 CF            01  FB 00091         CALLS   #1, PUT_LINE                      ;
                                            04 00096         RET                                       ; 0442

; Routine Size:  151 bytes,    Routine Base:  $CODE$ + 014A


;  329          0443  1
;  331             0444   1 %SBTTL  'bad$produce_report -- Generate report from detected bad sector file(s)'
;  332             0445   1 GLOBAL ROUTINE bad$produce_report : NOVALUE =
;  333             0446   1 !++
;  334             0447   1 !
;  335             0448   1 ! Functional Description:
;  336             0449   1 !
;  337             0450   1 !     This routine generates the report from the detected bad sector file(s).
;  338             0451   1 !
;  339             0452   1 ! Implicit Inputs:
;  340             0453   1 !
;  341             0454   1 !     Global data.
;  342             0455   1 !
;  343             0456   1 !--
;  344             0457   2 BEGIN
;  345             0458   2 new_page ();                                              ! Generate a new page.
;  346             0459   2 IF .bad$gl_context [ctx_v_ltdevice]
;  347             0460   2 THEN
;  348             0461   3     BEGIN
;  349             0462   3     list_last_track ();                                   ! Last track devices.
;  350             0463   3     put_line ();                                          ! Skip another line and
;  351             0464   3     format_line (bad$_lsttotbk,                           ! write the total number
;  352             0465   3                  bad$gq_devnam,                           ! of blocks on the device
;  353             0466   3                  .bad$gl_maxblock,                        ! followed by the number of
;  354             0467   3                  .mdbsf_count + .sdbsf_count);            ! bad blocks detected.
;  355             0468   3     END
;  356             0469   2 ELSE
;  357             0470   3     BEGIN
;  358             0471   3     list_non_lsttrk ();                                   ! Non last track devices.
;  359             0472   3     put_line ();                                          ! Skip another line and
;  360             0473   3     format_line (bad$_lsttotbk,                           ! write the total number
;  361             0474   3                  bad$gq_devnam,                           ! of blocks on the device
;  362             0475   3                  .bad$gl_maxblock,                        ! followed by the number of
;  363             0476   3                  .sdbsf_count);                           ! bad blocks detected.
;  364             0477   2     END;
;  365             0478   2
;  366             0479   2 IF .bad$gl_context [ctx_v_blk0bad]                        ! If block 0 is bad,
;  367             0480   2 THEN                                                      ! be sure to include the
;  368             0481   3     BEGIN                                                 ! warning in the report.
;  369             0482   3     put_line ();                                          ! Space it for visibility.
;  370             0483   3     format_line (bad$_blk0bad, bad$ga_device);            ! Write the warning.
;  371             0484   2     END;
;  372             0485   2
;  373             0486   2 put_line ();                                              ! Skip a line and write
;  374             0487   2 format_line (bad$_lststring, bad$ga_comnd_line);          ! the command line.
;  375             0488   2
;  376             0489   1 END;      ! of GLOBAL ROUTINE bad$produce_report



                                            0004 00000          .ENTRY  BAD$PRODUCE_REPORT, Save R2                          ; 0445
                     52      FF21     CF     9E 00002          MOVAB   PUT_LINE, R2                                          ;
                83   A2               00     FB 00007          CALLS   #0, NEW_PAGE                                          ; 0458
                     12      0000G    CF     E9 0000B          BLBC    BAD$GL_CONTEXT+1, 1$                                  ; 0459
             0000V   CF               00     FB 00010          CALLS   #0, LIST_LAST_TRACK                                   ; 0462
                     62               00     FB 00015          CALLS   #0, PUT_LINE                                          ; 0463
```

```
                      7E      0000'   CF      0000'   CF   C1 00018          ADDL3   SDBSF_COUNT, MDBSF_COUNT, -(SP)              ; 0467
                                                      0C   11 00020          BRB     2$                                           ; 0466
                              0000V   CF              00   FB 00022 1$:      CALLS   #0, LIST_NON_LSTTRK                          ; 0471
                                      62              00   FB 00027          CALLS   #0, PUT_LINE                                 ; 0472
                                              0000'   CF   DD 0002A          PUSHL   SDBSF_COUNT                                  ; 0476
                                              0000G   CF   DD 0002E 2$:      PUSHL   BAD$GL_MAXBLOCK                              ; 0475
                                              0000G   CF   9F 00032          PUSHAB  BAD$GQ_DEVNAM                                ; 0473
                                          00000000G   8F   DD 00036          PUSHL   #BAD$_LSTTOTBK                               ;
                                 42   A2              04   FB 0003C          CALLS   #4, FORMAT_LINE                              ;
                      11      0000G   CF              01   E1 00040          BBC     #1, BAD$GL_CONTEXT, 3$                       ; 0479
                                      62              00   FB 00046          CALLS   #0, PUT_LINE                                 ; 0482
                                              0000G   CF   9F 00049          PUSHAB  BAD$GA_DEVICE                                ; 0483
                                          00000000G   8F   DD 0004D          PUSHL   #BAD$_BLKOBAD                                ;
                                 42   A2              02   FB 00053          CALLS   #2, FORMAT_LINE                              ;
                                      62              00   FB 00057 3$:      CALLS   #0, PUT_LINE                                 ; 0486
                                              0000G   CF   9F 0005A          PUSHAB  BAD$GA_COMND_LINE                            ; 0487
                                          00000000G   8F   DD 0005E          PUSHL   #BAD$_LSTSTRING                              ;
                                 42   A2              02   FB 00064          CALLS   #2, FORMAT_LINE                              ;
                                                           04 00068          RET                                                  ; 0489

; Routine Size:  105 bytes,    Routine Base:  $CODE$ + 01E1

;  377          0490  1
;  379          0491  1 %SBTTL  'bad$close_files -- Close all open files'
;  380          0492  1 GLOBAL ROUTINE bad$close_files : NOVALUE =
;  381          0493  1 !++
;  382          0494  1 !
;  383          0495  1 ! Functional Description:
;  384          0496  1 !
;  385          0497  1 !     Close the output file here to trap any weird problems.
;  386          0498  1 !
;  387          0499  1 ! Implicit Inputs:
;  388          0500  1 !
;  389          0501  1 !     report_fab          the address of the report fab.
;  390          0502  1 !
;  391          0503  1 ! Side Effects:
;  392          0504  1 !
;  393          0505  1 !     Any problems in closing the output file will be reported via
;  394          0506  1 !     a SIGNAL.
;  395          0507  1 !
;  396          0508  1 !--
;  397          0509  2 BEGIN
;  398          0510  2
;  399          0511  2 report_fab [fab$l_ctx] = bad$_closeout;                  ! Pass error code.
;  400          0512  2 $CLOSE (FAB = report_fab, ERR = util$report_io_error);  ! Close the output file.
;  401          0513  2
;  402          0514  1 END;     ! of GLOBAL ROUTINE bad$close_files



                                                                          .EXTRN   SYS$CLOSE

                                                       0000 00000         .ENTRY   BAD$CLOSE_FILES, Save nothing                   ; 0492
                                0000'  CF 00000000G  8F   D0 00002         MOVL     #BAD$_CLOSEOUT, REPORT_FAB+24                   ; 0511
                                          00000000G  00   9F 0000B         PUSHAB   UTIL$REPORT_IO_ERROR                            ; 0512
                                              0000'  CF   9F 00011         PUSHAB   REPORT_FAB                                      ;
                         00000000G  00               02   FB 00015         CALLS    #2, SYS$CLOSE                                   ;
                                                          04 0001C         RET                                                      ; 0514

; Routine Size:  29 bytes,    Routine Base:  $CODE$ + 024A


;  403          0515  1
;  405        0516  1 %SBTTL  'list_last_track -- List the contents of both the MDBSF and SDBSF'
;  406        0517  1 ROUTINE list_last_track : NOVALUE =
;  407        0518  1 !++
;  408        0519  1 !
;  409        0520  1 ! Functional Description:
;  410        0521  1 !
;  411        0522  1 !     This routine will list each of the entries in the MDBSF and SDBSF.
;  412        0523  1 !     We call the routine "bad$cvt_ltk_long" to convert the entries into
;  413        0524  1 !     seperate SECTOR, TRACK, and CYLINDER values.  As we encounter each
;  414        0525  1 !     entry in the respective bad block file, we increment the respective
;  415        0526  1 !     counts.  The entries are listed line by line and only those entries
;  416        0527  1 !     with valid data are listed.
;  417        0528  1 !
;  418        0529  1 ! Implicit Inputs:
;  419        0530  1 !
;  420        0531  1 !     bad$ga_mdbsf          address of the MDBSF buffer.
;  421        0532  1 !     bad$ga_sdbsf          address of the SDBSF buffer.
;  422        0533  1 !
;  423        0534  1 !--
;  424        0535  2 BEGIN
;  425        0536  2 LOCAL
;  426        0537  2     cyl1,                                               ! Cylinder number of MDBSF entry.
;  427        0538  2     trk1,                                               ! Track    number of MDBSF enrty.
;  428        0539  2     sec1,                                               ! Sector   number of MDBSF entry.
;  429        0540  2     cyl2,                                               ! Cylinder number of SDBSF entry.
;  430        0541  2     trk2,                                               ! Track    number of SDBSF entry.
;  431        0542  2     sec2,                                               ! Sector   number of SDBSF entry.
;  432        0543  2     mdbsf_present,                                      ! We have an MDBSF entry.
;  433        0544  2     sdbsf_present;                                      ! We have an SDBSF entry.
;  434        0545  2
;  435        0546  2 mdbsf_count = sdbsf_count = 0;                          ! Initialize the counters.
;  436        0547  2 INCR entry FROM 2 TO bad$k_page_size / 4 - 1 DO
;  437        0548  3     BEGIN
;  438        0549  3     IF mdbsf_present = get_entry (.entry, .bad$ga_mdbsf, sec1, trk1, cyl1)
;  439        0550  3     THEN
;  440        0551  3         mdbsf_count = .mdbsf_count + 1;
;  441        0552  3
;  442        0553  3     IF sdbsf_present = get_entry (.entry, .bad$ga_sdbsf, sec2, trk2, cyl2)
;  443        0554  3     THEN
;  444        0555  3         sdbsf_count = .sdbsf_count + 1;
;  445        0556  3
;  446        0557  3     IF .mdbsf_present AND .sdbsf_present
;  447        0558  3     THEN
;  448        0559  3         format_line (bad$_lstmdbs3, .sec1, .trk1, .cyl1, .sec2, .trk2, .cyl2)
;  449        0560  3     ELSE
;  450        0561  4         BEGIN
;  451        0562  4         IF .mdbsf_present
;  452        0563  4         THEN
;  453        0564  4             format_line (bad$_lstmdbsf, .sec1, .trk1, .cyl1)
;  454        0565  4         ELSE
;  455        0566  4             IF .sdbsf_present
;  456        0567  4             THEN
;  457        0568  4                 format_line (bad$_lstsdbsf, .sec2, .trk2, .cyl2);
;  458        0569  3         END;
;  459        0570  3
;  460        0571  2     END;
;  461        0572  2
;   462        0573  2 !
;   463        0574  2 ! At the conclusion of listing the contents of the MDBSF and SDBSF,
;   464        0575  2 ! write the summary of number of bad sectors recorded.
;   465        0576  2 !
;   466        0577  2 put_line ();                                                    ! Skip a line.
;   467        0578  2 format_line (bad$_lstmdbs4, .mdbsf_count, .sdbsf_count);! Write the respective buffer counts.
;   468        0579  2
;   469        0580  1 END;       ! of ROUTINE list_last_track



                                          007C 00000 LIST_LAST_TRACK:
                                                              .WORD    Save R2,R3,R4,R5,R6                          ; 0517
              56      FEDD  CF  9E 00002          MOVAB    FORMAT_LINE, R6                              ;
              55      0000' CF  9E 00007          MOVAB    SDBSF_COUNT, R5                              ;
              5E              18  C2 0000C          SUBL2    #24, SP                                      ;
                              65  D4 0000F          CLRL     SDBSF_COUNT                                  ; 0546
                      F8  A5  D4 00011          CLRL     MDBSF_COUNT                                  ;
              52              02  D0 00014          MOVL     #2, ENTRY                                    ; 0547
                              5E  DD 00017 1$:      PUSHL    SP                                           ; 0549
                      08  AE  9F 00019          PUSHAB   TRK1                                         ;
                      10  AE  9F 0001C          PUSHAB   SEC1                                         ;
                   0000G  CF  DD 0001F          PUSHL    BAD$GA_MDBSF                                 ;
                              52  DD 00023          PUSHL    ENTRY                                        ;
       0000V  CF              05  FB 00025          CALLS    #5, GET_ENTRY                                ;
              54              50  D0 0002A          MOVL     R0, MDBSF_PRESENT                            ;
              03              54  E9 0002D          BLBC     MDBSF_PRESENT, 2$                            ;
                      F8  A5  D6 00030          INCL     MDBSF_COUNT                                  ; 0551
                      0C  AE  9F 00033 2$:      PUSHAB   CYL2                                         ; 0553
                      14  AE  9F 00036          PUSHAB   TRK2                                         ;
                      1C  AE  9F 00039          PUSHAB   SEC2                                         ;
                   0000G  CF  DD 0003C          PUSHL    BAD$GA_SDBSF                                 ;
                              52  DD 00040          PUSHL    ENTRY                                        ;
       0000V  CF              05  FB 00042          CALLS    #5, GET_ENTRY                                ;
              53              50  D0 00047          MOVL     R0, SDBSF_PRESENT                            ;
              02              53  E9 0004A          BLBC     SDBSF_PRESENT, 3$                            ;
                              65  D6 0004D          INCL     SDBSF_COUNT                                  ; 0555
              33              54  E9 0004F 3$:      BLBC     MDBSF_PRESENT, 5$                            ; 0557
              1D              53  E9 00052          BLBC     SDBSF_PRESENT, 4$                            ;
                      0C  AE  DD 00055          PUSHL    CYL2                                         ; 0559
                      14  AE  DD 00058          PUSHL    TRK2                                         ;
                      1C  AE  DD 0005B          PUSHL    SEC2                                         ;
                      0C  AE  DD 0005E          PUSHL    CYL1                                         ;
                      14  AE  DD 00061          PUSHL    TRK1                                         ;
                      1C  AE  DD 00064          PUSHL    SEC1                                         ;
               00000000G  8F  DD 00067          PUSHL    #BAD$_LSTMDBS3                               ;
              66              07  FB 0006D          CALLS    #7, FORMAT_LINE                              ;
                              28  11 00070          BRB      7$                                           ;
              10              54  E9 00072 4$:      BLBC     MDBSF_PRESENT, 5$                            ; 0562
                              6E  DD 00075          PUSHL    CYL1                                         ; 0564
                      08  AE  DD 00077          PUSHL    TRK1                                         ;
                      10  AE  DD 0007A          PUSHL    SEC1                                         ;
               00000000G  8F  DD 0007D          PUSHL    #BAD$_LSTMDBSF                               ;
                              12  11 00083          BRB      6$                                           ;
              12              53  E9 00085 5$:      BLBC     SDBSF_PRESENT, 7$                            ; 0566
```

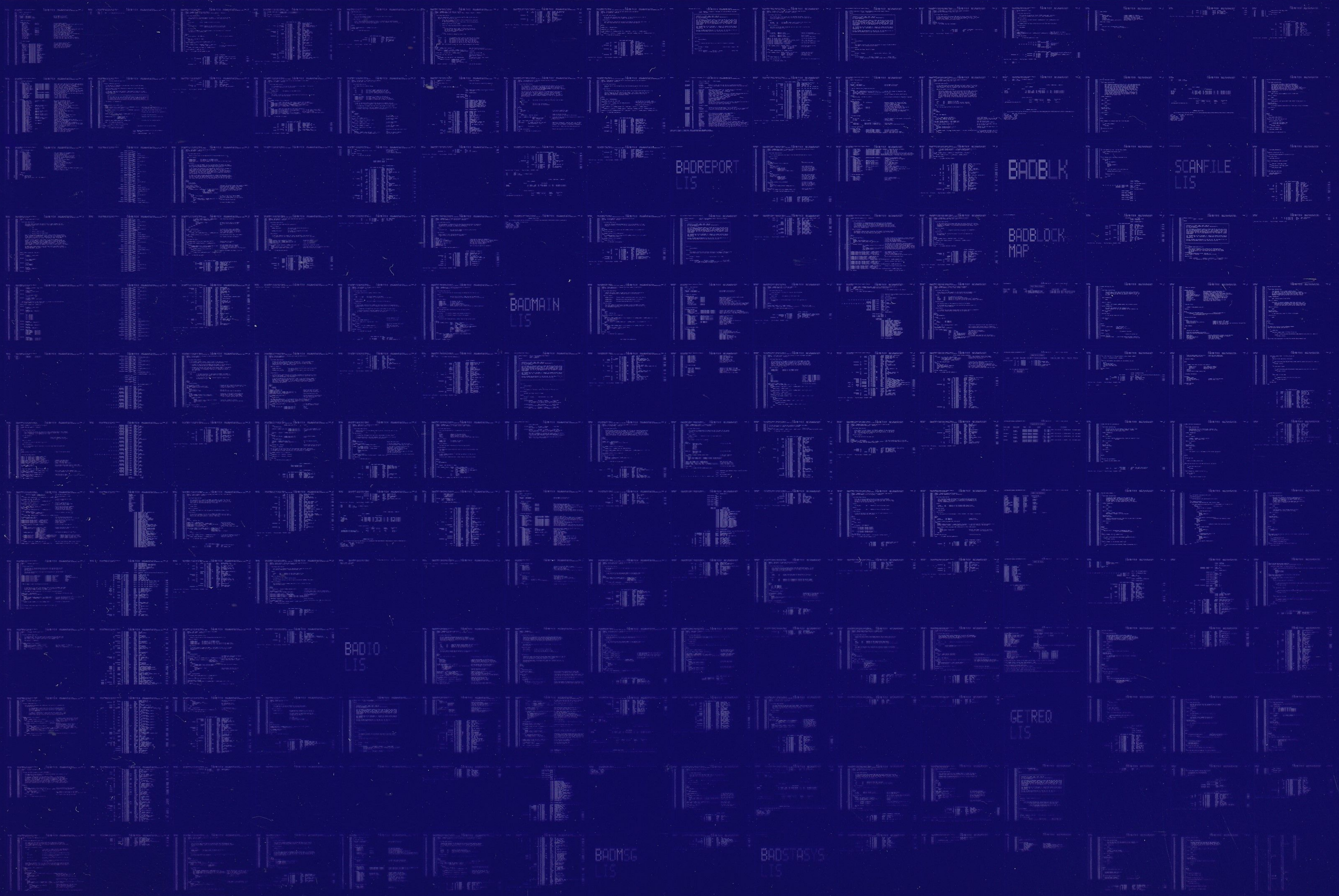
```
                                     0C   AE  DD 00088           PUSHL   CYL2                                     ; 0568
                                     14   AE  DD 0008B           PUSHL   TRK2                                     ;
                                     1C   AE  DD 0008E           PUSHL   SEC2                                     ;
                              00000000G   8F  DD 00091           PUSHL   #BAD$_LSTSDBSF                           ;
                          66         04   FB  00097 6$:          CALLS   #4, FORMAT_LINE                          ;
      FF73           52           01 0000007F  8F  F1 0009A 7$:  ACBL    #127, #1, ENTRY, 1$                      ; 0547
                         BE   A6     00   FB  000A4              CALLS   #0, PUT_LINE                             ; 0577
                                     65   DD  000A8              PUSHL   SDBSF_COUNT                              ; 0578
                                F8   A5   DD  000AA              PUSHL   MDBSF_COUNT                              ;
                              00000000G   8F  DD 000AD           PUSHL   #BAD$_LSTMDBS4                           ;
                          66         03   FB  000B3              CALLS   #3, FORMAT_LINE                          ;
                                          04  000B6              RET                                              ; 0580

; Routine Size:  183 bytes,     Routine Base:  $CODE$ + 0267

;  470          0581   1
;  472    0582  1 %SBTTL  'get_entry -- Retrieve the buffer entry'
;  473    0583  1 ROUTINE get_entry (index, buffer, sec, trk, cyl) =
;  474    0584  1 !++
;  475    0585  1 !
;  476    0586  1 ! Functional Description:
;  477    0587  1 !
;  478    0588  1 !     This routine extracts a single entry from the indicated buffer.  If the
;  479    0589  1 !     entry which it extracted was the end of file mark (-1), then we return
;  480    0590  1 !     FASLE, else if it was a valid entry, we will convert the entry into its
;  481    0591  1 !     constituent parts and return TRUE.
;  482    0592  1 !
;  483    0593  1 ! Inputs:
;  484    0594  1 !
;  485    0595  1 !     index       val      offset from the start of the buffer of the entry,
;  486    0596  1 !     buffer      adr      address of the buffer to extract an entry from.
;  487    0597  1 !
;  488    0598  1 ! Outputs:
;  489    0599  1 !
;  490    0600  1 !     sec         adr      address of a longword to receive the sector number.
;  491    0601  1 !     trk         adr      address of a longword to receive the track number.
;  492    0602  1 !     cyl         adr      address of a longword to recieve the cylinder number.
;  493    0603  1 !
;  494    0604  1 !--
;  495    0605  2 BEGIN
;  496    0606  2 MAP
;  497    0607  2     cyl : REF $BBLOCK,
;  498    0608  2     trk : REF $BBLOCK,
;  499    0609  2     sec : REF $BBLOCK,
;  500    0610  2     buffer : REF VECTOR [, LONG];
;  501    0611  2
;  502    0612  2 LOCAL
;  503    0613  2     value_present : BYTE;
;  504    0614  2
;  505    0615  2 value_present = FALSE;                                        ! Assume end of file has been reached.
;  506    0616  2 IF .buffer [.index] NEQ -1                                    ! Have we got a valid entry?
;  507    0617  2 THEN                                                         ! Yes -- convert the entry
;  508    0618  3     BEGIN                                                    ! into its constituent parts.
;  509    0619  3     bad$cvt_ltk_long (.buffer [.index], .sec, .trk, .cyl);
;  510    0620  3     value_present = TRUE;                                    ! Indicate its a valid entry.
;  511    0621  3     END;
;  512    0622  2
;  513    0623  2 RETURN .value_present;                                        ! Return with boolean indicator.
;  514    0624  2
;  515    0625  1 END;     ! of ROUTINE get_entry



                                              0004 00000 GET_ENTRY:
                                                         .WORD   Save R2                                   : 0583
                                     52   94 00002       CLRB    VALUE_PRESENT                             : 0615
                 50         04       AC   D0 00004       MOVL    INDEX, R0                                 : 0616
        FFFFFFFF 8F         08 BC40       D1 00008       CMPL    @BUFFER[R0], #-1                          :
                                     15   13 00011       BEQL    1$                                        :
                 7E         10       AC   7D 00013       MOVQ    TRK, -(SP)                                : 0619
                            0C       AC   DD 00017       PUSHL   SEC                                       :
```

```
                          08 BC40  DD 0001A        PUSHL   @BUFFER[R0]                              ;
            00000000G  00      04  FB 0001E        CALLS   #4, BAD$CVT_LTK_LONG                     ;
                       52      01  90 00025        MOVB    #1, VALUE_PRESENT                        ; 0620
                       50      52  9A 00028 1$:    MOVZBL  VALUE_PRESENT, R0                        ; 0623
                                   04 0002B        RET                                              ; 0625

; Routine Size:  44 bytes,    Routine Base:  $CODE$ + 031E


;  516          0626  1
;  518      0627  1 %SBTTL  'list_non_lsttrk -- List the contents of the SDBSF'
;  519      0628  1 ROUTINE list_non_lsttrk : NOVALUE =
;  520      0629  1 !++
;  521      0630  1 !
;  522      0631  1 ! Functional Description:
;  523      0632  1 !
;  524      0633  1 !       List the contents of the non last track SDBSF.  Here we have a count of
;  525      0634  1 !       the number of used bad block descriptors and will iteratively extract
;  526      0635  1 !       and convert the descriptor into our output format.  The completion will
;  527      0636  1 !       simply list the total number of bad blocks recorded.
;  528      0637  1 !
;  529      0638  1 ! Implicit Inputs:
;  530      0639  1 !
;  531      0640  1 !       bad$ga_sdbsf            address of the SDBSF buffer.
;  532      0641  1 !
;  533      0642  1 !--
;  534      0643  2 BEGIN
;  535      0644  2 LOCAL
;  536      0645  2      lbn,
;  537      0646  2      count : BYTE;
;  538      0647  2
;  539      0648  2 sdbsf_count = 0;                                            ! Initialize counter.
;  540      0649  2 INCR entry FROM .bad$ga_sdbsf + nlt_k_headersiz             ! Extract the used
;  541      0650  2            TO .bad$ga_sdbsf + nlt_k_headersiz +             ! bad block descriptors
;  542      0651  3               (.bad$ga_sdbsf [nlt_b_usedbbdsc] * 2 - 4)     ! and convert the entrys
;  543      0652  2            BY 4 DO                                          ! into longwords, which
;  544      0653  3      BEGIN                                                  ! will be used in the
;  545      0654  3      bad$cvt_nlt_long (..entry, count, lbn);                ! listing.  Keep a count
;  546      0655  3      sdbsf_count = .sdbsf_count + .count + 1;               ! of the number of bad
;  547      0656  3      format_line (bad$_lstsdbs3, .lbn, .count);             ! blocks as we write the
;  548      0657  2      END;                                                   ! listing.
;  549      0658  2
;  550      0659  1 END;     ! of ROUTINE list_non_lsttrk



                                            001C 00000 LIST_NON_LSTTRK:
                                                               .WORD   Save R2,R3,R4                        ; 0628
                        54      0000'   CF   9E 00002          MOVAB   SDBSF_COUNT, R4                      ;
                        5E              08   C2 00007          SUBL2   #8, SP                               ;
                                        64   D4 0000A          CLRL    SDBSF_COUNT                          ; 0648
                        52      0000G   CF   D0 0000C          MOVL    BAD$GA_SDBSF, R2                     ; 0649
                        50         02   A2   9A 00011          MOVZBL  2(R2), R0                            ; 0651
                        53            6240   3E 00015          MOVAW   (R2)[R0], R3                         ; 0650
                                        2B   11 00019          BRB     2$                                   ; 0649
                                        5E   DD 0001B 1$:      PUSHL   SP                                   ; 0654
                                   08   AE   9F 0001D          PUSHAB  COUNT                                ;
                                        62   DD 00020          PUSHL   (ENTRY)                              ;
                 00000000G      00      03   FB 00022          CALLS   #3, BAD$CVT_NLT_LONG                 ;
                        50         04   AE   9A 00029          MOVZBL  COUNT, R0                            ; 0655
                        50              64   C0 0002D          ADDL2   SDBSF_COUNT, R0                      ;
                        64         01   A0   9E 00030          MOVAB   1(R0), SDBSF_COUNT                   ;
                        7E         04   AE   9A 00034          MOVZBL  COUNT, -(SP)                         ; 0656
                                   04   AE   DD 00038          PUSHL   LBN                                  ;
                              00000000G 8F   DD 0003B          PUSHL   #BAD$_LSTSDBS3                       ;
```

```
                         FDBA    CF                 03  FB 00041            CALLS   #3, FORMAT_LINE                          ;
        FFCF            52              04          53  F1 00046 2$:        ACBL    R3, #4, ENTRY, 1$                        ; 0649
                                                        04 0004C            RET                                              ; 0659

; Routine Size:  77 bytes,    Routine Base:  $CODE$ + 034A


;  551          0660  1
;  552          0661  1 END
;  553          0662  0 ELUDOM



                                                                                  .EXTRN  LIB$SIGNAL

;                                         PSECT SUMMARY
;
;       Name                        Bytes                          Attributes
;
;  $OWN$                              164  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                              20  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                             919  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                                   Library Statistics
;
;                                                 -------- Symbols --------         Pages        Processing
;       File                                      Total     Loaded    Percent       Mapped       Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1                  18619         80          0       1000           00:01.4



;                                          COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:BADREPORT/OBJ=OBJ$:BADREPORT MSRC$:BADREPORT/UPDATE=(ENH$:BADREPORT)

; Size:           919 code + 184 data bytes
; Run Time:          00:13.2
; Elapsed Time:      00:58.0
; Lines/CPU Min:       3013
; Lexemes/CPU-Min:   30578
; Memory Used:   143 pages
; Compilation Complete
```

0018 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



BADREPORT
LIS

BADBLK

SCANFILE
LIS

BADBLOCK
MAP

BADMAIN
LIS

BADIO
LIS

GETREQ
LIS

BADMSG
LIS

BADSTASYS
LIS